AD-A253 344

In Press: Simulation

(2)

# Simulating Biological Vision with Hybrid Neural Networks

DTIC
ELECTE
JUL 2 8 1992
S C D

Paul Sajda and Leif H. Finkel

Department of Bioengineering and
Institute of Neurological Sciences
University of Pennsylvania
Philadelphia, PA. 19104-6392

[illegible] STATEMENT A
[illegible] release;
Distribution Unlimited



92-20021

92 7 24 032

DTIC QUALITY INSPECTED 3

Accession For
NTIS GRA&I
DTIC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

Statement A per telecon Harold Hawkins
ONR/Code 1142
Arlington, VA 22217-5000
NWW 7/27/92

## Abstract

We present an example of how vision systems can be modeled and designed by integrating a top–down computationally–based approach with a bottom–up biologically–motivated architecture. The specific visual processing task we address is occlusion–based object segmentation–the discrimination of objects using cues derived from object interposition. We construct a model of object segmentation using *hybrid neural networks*–distributed parallel systems consisting of neural units modeled at different levels of abstraction. We show that such networks are particularly useful for systems which can be modeled using the combined top–down/bottom–up approach. Our hybrid model is capable of discriminating objects and stratifying them in relative depth. In addition, our system can account for several classes of human perceptual phenomena, such as illusory contours. We conclude that hybrid systems serve as a powerful paradigm for understanding the information processing strategies of biological vision and for constructing artificial vision–based applications.

**Keywords:** vision, neural networks, hybrid systems, biologically–based modeling, human perception.

## I. Introduction

Research in vision has been largely dominated by two philosophies (Marr, 1982). One is the *top–down* approach, advocated by those in computer and machine vision, whose primary aim is developing task-specific visual processing systems. The top–down methodology emphasizes computational theory in the development of functional systems. The *bottom–up* philosophy proposes that the biological implementation must be considered when studying visual processing. Since the most efficient and robust "vision machine" to date, and the yardstick against which all artificial vision systems are usually measured, is the human visual system, one should not neglect the architecture established by evolution. The bottom–up advocates claim that "reverse engineering", or using the low–level implementation to determine the system's functional behavior, is the key to understanding and designing both

biological and artificial vision systems.

Recently a field, known as computational neuroscience (Sejnowski et al., 1988), has emerged which realizes the importance of applying both of these approaches simultaneously. By addressing the problem with regard to both top–down and bottom–up philosophies one is better able to circumvent the shortcomings inherent to the individual approaches. For example, an understanding of the neurophysiology and biological circuitry offers a powerful constraint on the system's function. However, this data is limited and the approach is not feasible when considering complex high–level processing. In these cases, one might also consider computational theories and evidence from visual psychology in formulating a working model of visual processing.

In this paper we present a specific example of how a bilateral approach, incorporating aspects from top–down and bottom–up methodologies, can be used to model object segmentation and depth processing. We show how traditional biologically-motivated neural networks can be integrated with networks consisting of functionally complex units to create a *hybrid neural network* model. Many elements of the model are consistent with known neurophysiology, while others rely on computational methods and psychological data to arrive at their design. We will demonstrate with simulations how our hybrid system functions to segment objects and stratify them in depth. In addition, we will test our model against human psychophysics, illustrating how it can account for perceptual phenomena, such as illusory contours (Kanizsa, 1979).

## II. Overview of the model

Object segmentation is a categorization process aimed at grouping regions of an image into meaningful representations. It occurs at an intermediate stage of the transformation between 2-D image intensities and visual recognition, and in general, depends upon information from multiple visual modalities (such as color, motion, texture and shading). To simplify the problem, we have restricted ourselves to segmentation based solely on occlusion

relationships. Occlusion is a consequence of projecting a three dimensional scene onto a two dimensional receptor array, and results from the interposition of objects with each other or the background. In a typical visual scene, multiple objects occlude one another, creating a perceptual dilemma–to which of the two overlapping surfaces does the common border belong? If the border is, in fact, an occlusion border, then it belongs to the occluding object. This identification results in a stratification of the two objects in depth and a de facto discrimination of the objects. For example, consider the case of a horse behind a tree. We perceive the tree as being closer than the horse, and in addition, the two "halves" of the horse created by the occlusion are linked into one object. Thus, though occlusion may isolate different regions of an object, our visual system is able to overcome this difficulty and provide a consistent and coherent representation of the scene's constituent objects.

A system must identify if an occlusion relationship exists if it is to accurately segment an image and determine relative depth. Since occlusion implies discontinuous depth, one can conclude that discontinuities in the image provide important occlusion information. Given that an object always occludes its background, all objects possess an *occluding contour* (Marr, 1977) in their two dimensional image. An occluding contour is a closed curve which "outlines" an object's silhouette. Though an occluding contour signals occlusion with the background, it alone gives little information about depth relationships between objects. In the two dimensional image there are a number of cues which imply object interposition. Figure 1 illustrates the primary cues for object occlusion. The strongest cue is the *T-junction.* At a T–junction the contours of occluding and occluded objects meet. T–junctions have long been recognized as important cues for scene segmentation (Guzman, 1968). Two other cues to occlusion are *concavities* and *surrounded contours.* Objects at different depths have overlapping two dimensional images, creating concavities in the occluding contours of the objects. The presence of concavities can therefore serve as an indicator for occlusion. Another occlusion cue occurs when a smaller object is in front of a larger object but no T–junctions are created. In this case the smaller object is completely surround by the larger

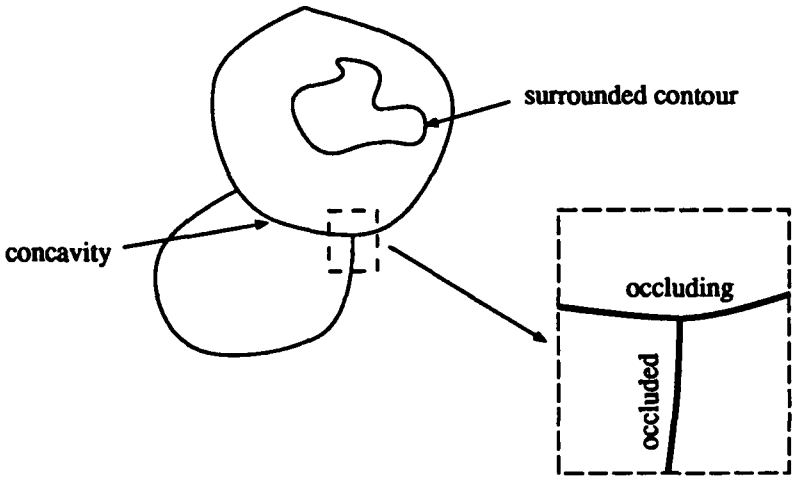



Figure 1: Cues to object occlusion. T–junctions (shown in inset) signal a local discontinuity between occluded and occluding contours. Concavities and surrounded contours suggest occlusion, but are not as reliable indicators as T–junctions.

object's occluding contour. This surround condition can be interpreted as a cue to occlusion (Koffka, 1935), with the smaller object perceived as being closer to the viewer. However, since objects often contain concavities or surrounded contours (for example an annulus) as part of their structure, neither concavities nor surrounded contours are as strong a cue to occlusion as T–junctions.

Our model identifies and uses these occlusion cues to segment the two dimensional image and determine the relative depth of objects. A schematic of information flow in the model is shown in figure 2. The system is organized into four main categories of processing; *feature extraction, segmentation and binding, depth processing* and *completion processing.* These in turn are divided into subcategories which represent functions performed by particular networks in the system.

The first stage of the model discriminates low–level features. Edges, oriented lines, line endings and junctions are detected by networks of units selective for these image attributes. The next stage involves segmentation and binding, which includes grouping features into *proto–objects.* We define proto–objects as *bounded, simply connected and spatially continuous surfaces with associated attributes, such as depth, color, and texture.* Proto–objects

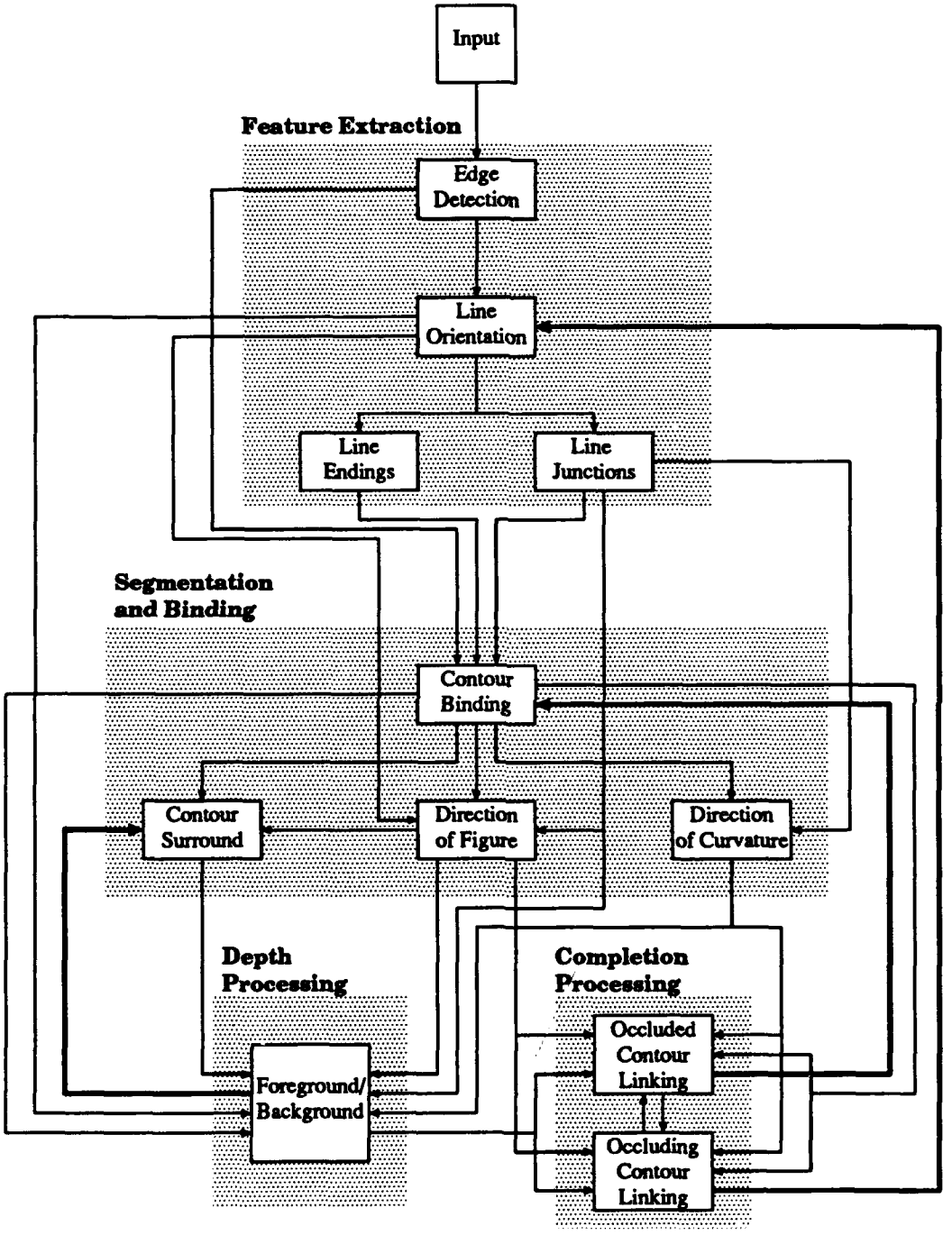



Figure 2: Organization and information flow in our model of object segmentation. Shown are the four primary stages of processing in the system and their associated subprocesses. Each subprocess is simulated in the model by either one or several networks. (Feedforward information flow is shown with thin arrows, with thick arrows representing feedback.)

are the precursors of objects, since feedback from completion processing (see figure 2) can group one or more proto–objects into a single object. The third stage of the model involves completing occluded and occluding contours. In our previous example, the two regions of the horse separated by the occluding tree are perceptually linked to form a single object.[1] This stage of the model includes mechanisms for linking unoccluded portions of proto–objects. In addition, occluding contours may be incomplete. For example, the intensity gradient between the tree and the horse may be small over a region of the image, and therefore no edge discontinuity is detectable. Thus, unambiguous edges are linked within the model to form continuous closed occluding contours. The final stage of the model is concerned with depth processing. Here a cooperative/competitive mechanism uses occlusion cues, identified in the earlier stages of processing, as "forces" for "pushing" objects into different relative depths. Depth in our model is represented in a distributed fashion between units in *foreground* and *background* networks. This distributed representation of depth is consistent with how disparity is represented in visual cortex (Poggio et al., 1988; Lehky and Sejnowski, 1990).

## III. Implementation of the model

To illustrate the hybrid nature of our model we will focus on the design, implementation, and simulation of two specific networks. The complete model is constructed and analyzed using the NEXUS Neural Simulator (Sajda and Finkel, 1992). NEXUS is an interactive simulation package designed for modeling multiple interconnected neural maps. Present simulations consist of 42 topographically organized interconnected networks, each containing an array of 64x64 units. A particularly novel feature of NEXUS is its ability to integrate different network paradigm's into a hybrid network model. One advantage of hybrid networks is that they allow one to simulate an entire system with units modeled at several different levels of abstraction. NEXUS incorporates this variability in the level of abstraction of

[1] Within our model, the two"halves" of the horse would be classified as proto–objects.

the individual units by defining PGN (Programmable Generalized Neural) units, which are capable of executing arbitrary functions or algorithms. PGN units are particularly useful in situations where intensive computations are performed but the anatomical and physiological details of the operation are unknown. In addition, instead of representing the function of a single neuron, PGN units can be used to model large cell assemblies and neural circuits. For example, a fully–connected winner–take–all circuit can be replaced by a single PGN unit (see Sajda and Finkel, 1992 for an example of how PGN units can be used in this fashion). Finally, PGN units allow the user, faced with finite computational resources, to manipulate the space–time trade–off [2] by reducing explicit network connectivity in favor of increased algorithmic complexity and processing time.

Our occlusion–based object segmentation system is a hybrid network model consisting of two classes of units. One set of units explicitly incorporates the known neurophysiology and connectivity seen in visual areas V1 and V2 into its network circuitry. These units are used in the early stages of the model (feature extraction and selected networks in the segmentation and binding stage). Later stages consist of the second class of unit (PGN units). In these cases, the lack of detailed neurophysiological data prohibits an accurate model of the neural architecture. However, we have included empirical evidence concerning the collective network properties of the visual cortex, such as oscillations and phase–locking (Gray and Singer, 1989; Eckhorn et al., 1988) and corticocortical connectivity (Rockland and Lund, 1982), into the functional behavior of the PGN units. Thus, though the entire system is not modeled at the level of the individual neuron, all stages are biologically–motivated.

In the following sections we will discuss two different networks in our hybrid system, each modeled at a different level of abstraction. The first functions in determining the local orientation of lines and contours, and is constructed using well–known physiological data. The second network uses PGN units to identify the *direction of figure*, or which side of the

[2] Tradeoff of CPU execution time and system memory.

occluding contour is the inside of the object–a problem often formulated as distinguishing figure from ground (Mumford et al., 1987, Sejnowski and Hinton, 1987). Finally, we will present simulations of how our hybrid model responds to visual stimuli and compare these responses to human visual perception.

## Orientation network

Neurons selective for orientation have been found in many areas of the visual cortex (Hubel and Wiesel, 1962), with cortical areas V1 and V2 having a majority of cells tuned to particular orientations. The functional behavior of these cells arises from the nature of their receptive fields. A receptive field describes a cell's response when selected areas of the receptor array (in this case the retina) are stimulated. The shape of a receptive field, and therefore the functional properties of the neuron, can be controlled by regulating the cell's connection pattern. For example, a center–surround receptive field approximating the second derivative operator, useful as an edge detector, can be constructed by creating a small central excitatory set of connections (connections with positive weights) and a larger surrounding set of inhibitory connections (negative weights). We call this pattern of weights the *connection field* of the cell. Figure 3A shows one type of connection field used for the orientation selective units in our model. Receptor activity at a given location is multiplied by the connection field value, and then summed to form the input voltage to the cell. A sigmoidal activation function (see figure 3B), representing the rectification and saturation properties of biological neurons, determines the cell's firing rate given its input voltage. The result is a unit which responds maximally to a particular orientation. Figure 3C shows a tuning curve for a cell having a maximum response for a $0^o$ line. Several orientation networks are used in the model, each having a different optimal stimulus orientation. The output of these units contributes to the "form" definition or internal representation of the proto–objects and serves as an approximation to the local tangent of the contour.

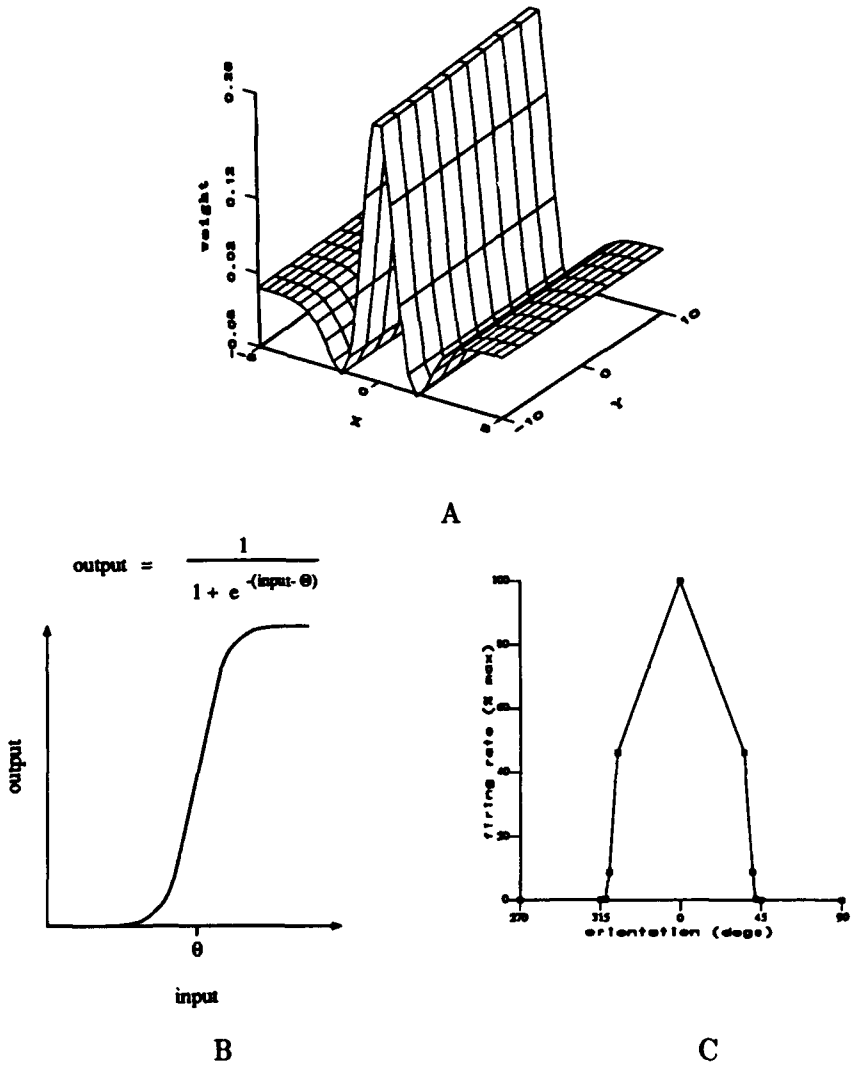


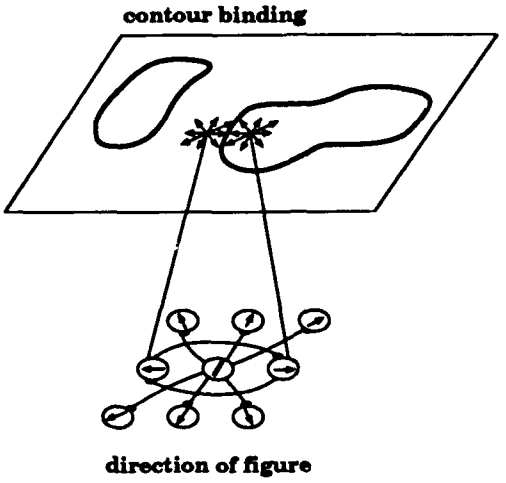

Figure 3: **A** Connection field for orientation selective units. A unit having this particular connection pattern would be selective for line segments parallel to the y-axis (0° or vertical orientation). **B** Sigmoidal activation function for an orientation selective units. **C** Tuning curve for a unit having a maximum response for a vertical (0°) line segment. The width of the tuning curve can be controlled by modifying either the connection field or the activation function.

## Direction of figure network

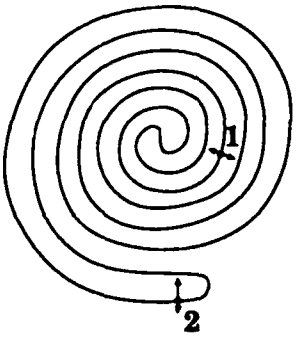
Though networks in the feature extraction stage of the model can identify edges and determine orientation and curvature of contours, they are not sufficient for segmenting the image into its constituent proto–objects. The remaining part of the problem requires that the surface of the object be identified. The task of the direction of figure network is to determine which side of the contour is the "inside" (surface) of the object and which is the "outside" (background). The problem can be restated as determining which region "owns" the contour (Koffka, 1935; Nakayama and Shimojo, 1990; Finkel and Sajda, 1992).

A schematic of a neural mechanism for computing the direction of figure is shown in figure 4A. The function of this circuit is based on the following simple observation. Suppose a unit projects its dendrites (connections) in a stellate configuration and that these dendrites are activated by units responding to a particular contour. Then if a given unit is inside a contour, more of its dendrites will be activated than if it is outside the contour. A winner–take–all mechanism between two such units will determine which is more strongly activated, and hence which is the inside of the object. As shown in figure 4A it is advantageous to limit this competition to the two units which are located at positions directly perpendicular to the local orientation (tangent) of the contour. It is important to note that this mechanism is consistent with human perception in that it will fail to identify the correct direction of figure for selected cases (see figure 4B).

Explicitly modeling the connectivity suggested in figure 4 requires a tremendous number of connections. Each unit in the direction of figure network receives input which spans the area of the contour binding network in eight different directions. For example, a direction of figure unit located at the center of the network would receive input from $8(\frac{\sqrt{N}}{2})$ contour binding units, where $N$ is the number of units in a network. Units at edges or corners would have fewer inputs ($3\sqrt{N}$) due to truncation of connections at network borders. Therefore each unit in the direction of figure network requires between $3\sqrt{N}$ and $4\sqrt{N}$ connections with units in the contour binding network. The total number of connections necessary




A



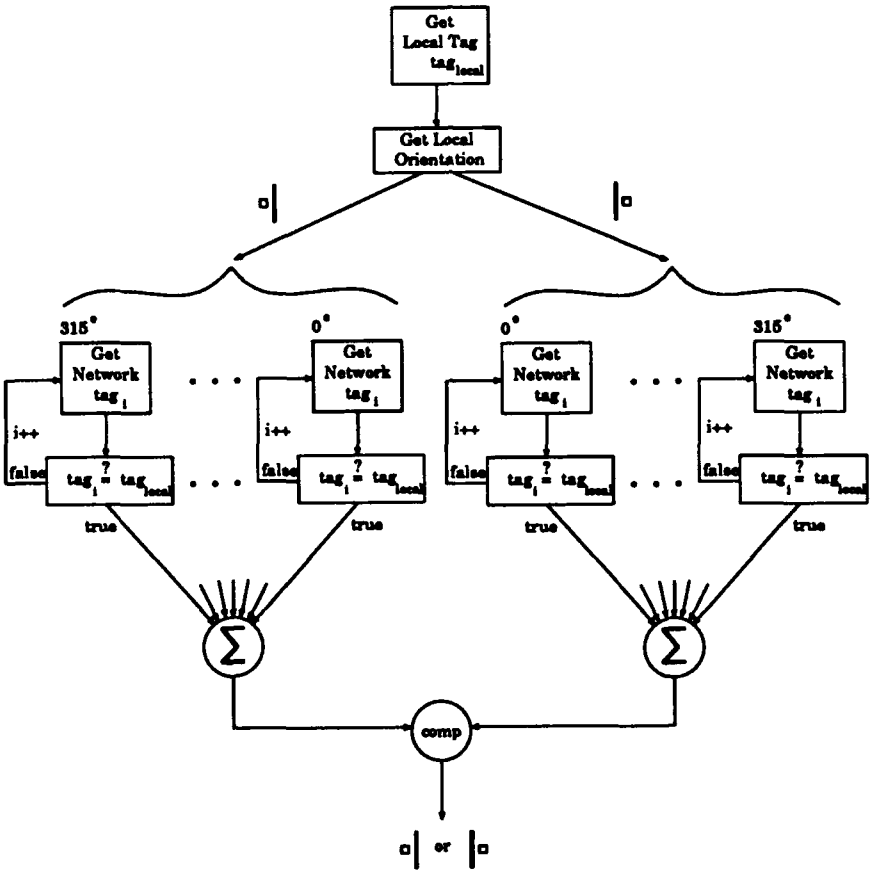

B

Figure 4: **A** Neural circuit for determining direction of figure (inside vs. outside), with a hypothetical input stimulus consists of two closed contours (bold curves). The central unit of 3x3 array (shown below) determines the local orientation of the contour using the output from the orientation networks. Surrounding units represent possible directions (indicated by arrows) of the inside of the figure relative to the contour. All surrounding units are inhibited (black circles) except for two units located perpendicular to the local orientation of the contour. Units receive inputs from the contour binding network via dendrites (connections) that spread out in a stellate configuration, as indicated by the clustered arrows (dendrites extend over long distances in the network). Units inside the object will receive more inputs than those units outside. The two uninhibited units compete using a winner–take–all mechanism. Note that inputs from separate objects are not confused due to the tags generated in the contour binding map. **B** An example of a stimulus where our proposed neural circuit cannot correctly determine direction of figure for the entire object. At point 1, there will be the same number of inputs from sides *a* and *b*, leaving the direction of figure ambiguous. This is consistent with human perception since subjects have difficulty instantaneously identifying the inside of the spiral.

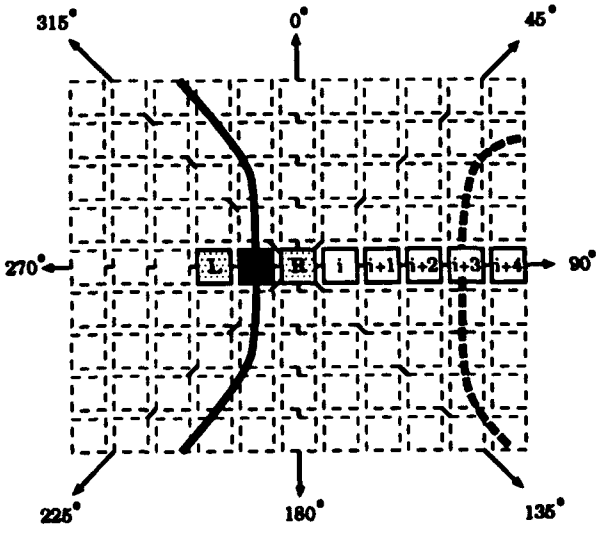
for a 64x64 network of direction of figure units would therefore be between 780,000 and 1,050,000! This number excludes the connections within the direction of figure network itself. In addition, these connections are not simple multiplicative coefficients. The activity in the contour binding network represents a "tag" for the individual occluding contours defining the proto–objects. Units which lie on the same contour are bound together with this common tag[3]. The connectivity between the contour binding and direction of figure networks must therefore be selective for particular tags, so that activity is summed only for a single occluding contour. This requires either an increase in the number of connections and the complexity of the supporting neural circuitry or connections which are functionally more complex than simple weighting coefficients.

Instead of explicitly establishing the connectivity, we model the function of the direction of figure network using PGN units. The algorithm we use is shown in figure 5A and the spatial layout of a section of the direction of figure network is shown in figure 5B. Every unit in the direction of figure network executes the algorithm in figure 5A. The functional blocks represent calls to procedures which allow a unit to retrieve the activity of a specific unit in a network given its spatial location. The regular topology of the model enables the system to trade–off the memory required for explicit connectivity in favor of increased execution time required for the calculation of the location of the input cell.

[3]It has been suggested that the biological substrate for such a binding mechanism may be cortical oscillations or phase–locking (Gray and Singer, 1989; Eckhorn et al., 1988).


Get Local Tag $tag_{local}$
Get Local Orientation
315°
0°
0°
315°
Get Network $tag_i$
i++
false
$tag_i \stackrel{?}{=} tag_{local}$
true
Σ
comp
or

A


315°
0°
45°
270°
90°
225°
180°
135°
L
R
i
i+1
i+2
i+3
i+4

B

Figure 5: *(previous page)* **A** Flowchart of the algorithm performed by PGN units in the direction of figure network. The units use procedure calls to determine activity levels in different networks (see text for discussion). **B** Illustration showing the layout of units in the direction of figure network. The unit for which the direction of figure is to be determined is shown in dark–gray, and the corresponding contour is shown as a solid black curve. The local orientation and contour binding tag is determined for the dark–gray unit, resulting in the inhibition of all unit except the two perpendicular to the local orientation (light–gray). Both these units sum activity in eight different directions by traversing through the network. For example, shown is the sequence of unit locations checked by the light–gray unit lying on the right side of the contour and in the 90° direction. (Dashed dark line represents location of a second contour segment intersecting $i + 3$.)

The first function in the algorithm gets the activity in the contour binding network representing the tag of the local segment of contour. For example, in figure 5B, the contour binding tag is determined for the dark–gray unit. Next the local orientation is identified using the activities from the orientation selective networks as inputs. The algorithm then forks into two different paths so that the summed activity on both sides of the contour can be determined separately. In figure 5B, this is shown by the two light-gray units perpendicular to the local orientation of the contour. The activity on either side of the contour is computed by traversing the network in eight different directions. The first step involves identifying the contour binding tag of the *ith* unit along the current direction. If this tag is equal to the local tag then the activity of the unit representing the particular contour is increased. If the tags are not equal then the system continues to traverse the network in the current direction. Figure 5B illustrates how the network is traversed for a unit on the left side of the contour and in a direction of 90°. Since the $i$, $i + 1$, and $i + 2$ units are not activated by a contour, they do not possess a tag. However, there is activity at the location of the $i + 3$ unit. If the contour binding tag at this position is equal to the local contour tag of the dark–gray unit then the activity of the left light-gray unit is increased, else the system continues to traverse the network in the 90° direction. Once all directions are traversed, the total activity for the two units on each side of the contour is compared. The side which has the greatest activity represents the local direction of figure–

the local direction of the inside of the object and therefore defines the region which "owns" the contour.

## IV. Simulation results

We present results from simulations which illustrate the ability of this hybrid system to discriminate objects. Figure 6 shows a scene that was presented to the system. The low–level networks detect edges, line orientation, terminations and junctions present in the scene. Figure 6A displays the activity in the contour binding network, representing the tags assigned to the different scene elements. Each box represents elements having a common tag, different boxes represent different tags, and the ordering of the boxes is arbitrary. On the first cycle discontinuous elements, such as the two regions of the horse, have separate tags. Feedback from the completion networks links these contours so that after the second cycle the contours defining the horse have the same tag.

The output of the direction of figure network for this particular stimulus is shown in figure 6B. The direction of the arrows indicates the direction of figure determined by the network. A small portion of the network is enlarged to better illustrate the system's performance. Note that the system correctly determines that the region representing the surface of the tree "owns" the vertical contour, while the surrounding contour is "owned" by the region of the horse.

T-junctions, such as those between the horse and the tree, force the various objects into different depth planes. The result of this process is shown in figure 6C, which plots the firing rate (as a percent of maximum) of units in the foreground network. The actual depth value determined for each object is somewhat arbitrary, and can vary depending upon minor changes in the scene–the system is designed to achieve the correct relative ordering, not absolute depth. In addition there is no way to determine the relative depth between the house and sun because they bear no occlusion relationship to each other. This conforms with human perception, e.g., the sun and the moon appear about the same distance away.

The hybrid system thus appears to process occlusion information in a manner similar to human perception.

The second simulation illustrates that the system displays a response consistent with human responses to illusory stimuli. Figure 7 shows a stimulus known as the Kanizsa square (Kanizsa, 1979). Human subjects typically perceive a white square occluding four black discs and a wireframe square. This perception is somewhat surprising given that the stimulus can just as easily be interpreted as four black "pacman-like" shapes and four angular line segments. Some have suggested that the perception of these illusions may arise from artificially arranged occlusion cues (Gregory, 1972).

Figure 7A shows the output of the direction of figure network after one and three cycles of activity. The large display shows that the surfaces of the objects (the discs, occluded and occluding squares) are correctly identified by the network after the third cycle. The two insets show an enlarged area of the network for both the first and third cycle. At first the system identifies the "L"–shaped mouth of the pacman as belonging to the disc, as illustrated by the direction of figure arrows. After the third cycle the "L"-shaped edge is identified as belonging to the occluding illusory square. This change in ownership of the edge results from the identification of occlusion–the edge has been identified as an occlusion border. Figure 7B displays the firing rate of units in the *foreground* depth map (as in figure 6C), thus showing that the system discriminates relative depth of the constituent objects.

# V. Conclusion

We have presented a hybrid system for occlusion–based object segmentation which builds upon data from several fields, including neurophysiology (Peterhans and von der Heydt, 1989; von der Heydt and Peterhans, 1989), neural computation (Ullman, 1976; Marr, 1982; Grossberg and Mingolla, 1985), psychophysics and psychology (Kanizsa, 1979; Nakayama, 1990) and computer vision (Rosenfeld, 1988; Fisher, 1989; Aloimonos and Shulman, 1989). In particular, we have emphasized the hybrid nature of our system by focusing on two

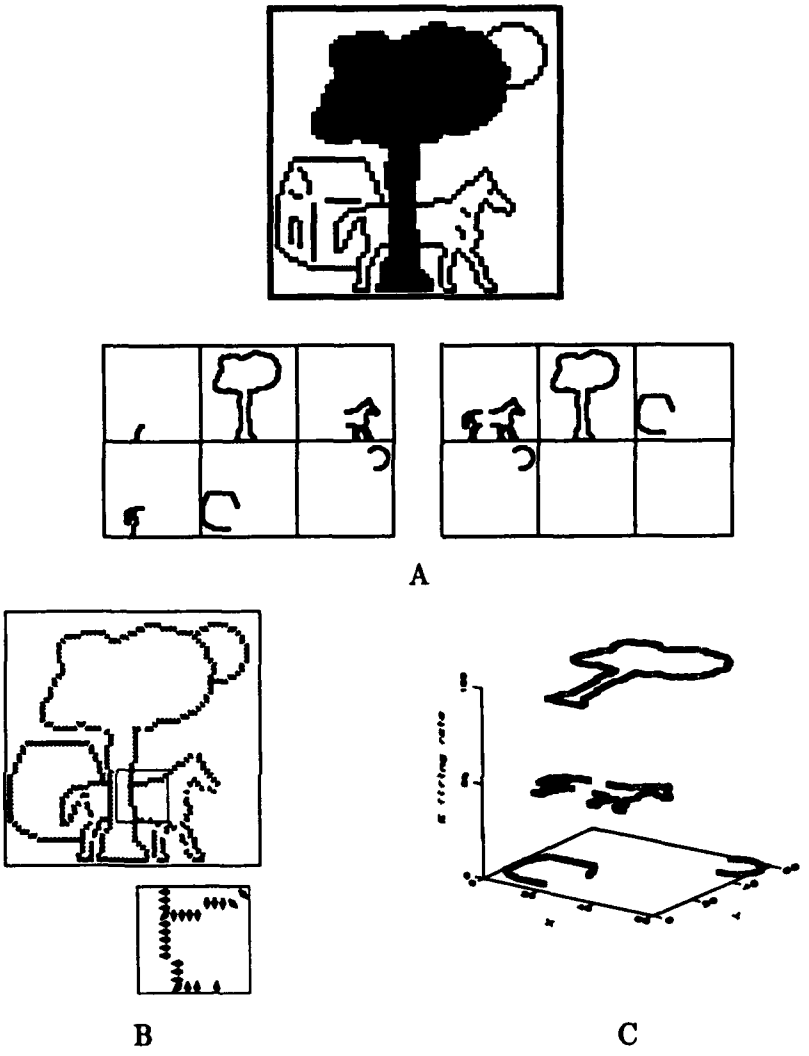



Figure 6: Object segmentation and stratification in depth. Top panel shows a 64x64 stimulus presented to the system. **A** Spatial histogram of the contour binding tags (each box shows a unit with a common tag, different boxes represent different tags, and the ordering of the boxes is arbitrary). Initial tags are shown on the left and the tags after two cycles are shown on the right. Note that the linking of occluded contours has transformed proto–objects into objects (the two sides of the horse have been linked to form a single object). **B** Output of the direction of figure network after two cycles. Inset shows a magnified view of the output of the direction of figure network for a local section of the image. Note that the system correctly assigns "ownership" of the vertical contour to the region of the tree, not the region of the horse. **C** Relative depth of objects in the scene as determined by the system. Plot of activity (% maximum) of units in the foreground network after two interactions. Points with higher activity are "perceived" as being closer to the viewer.

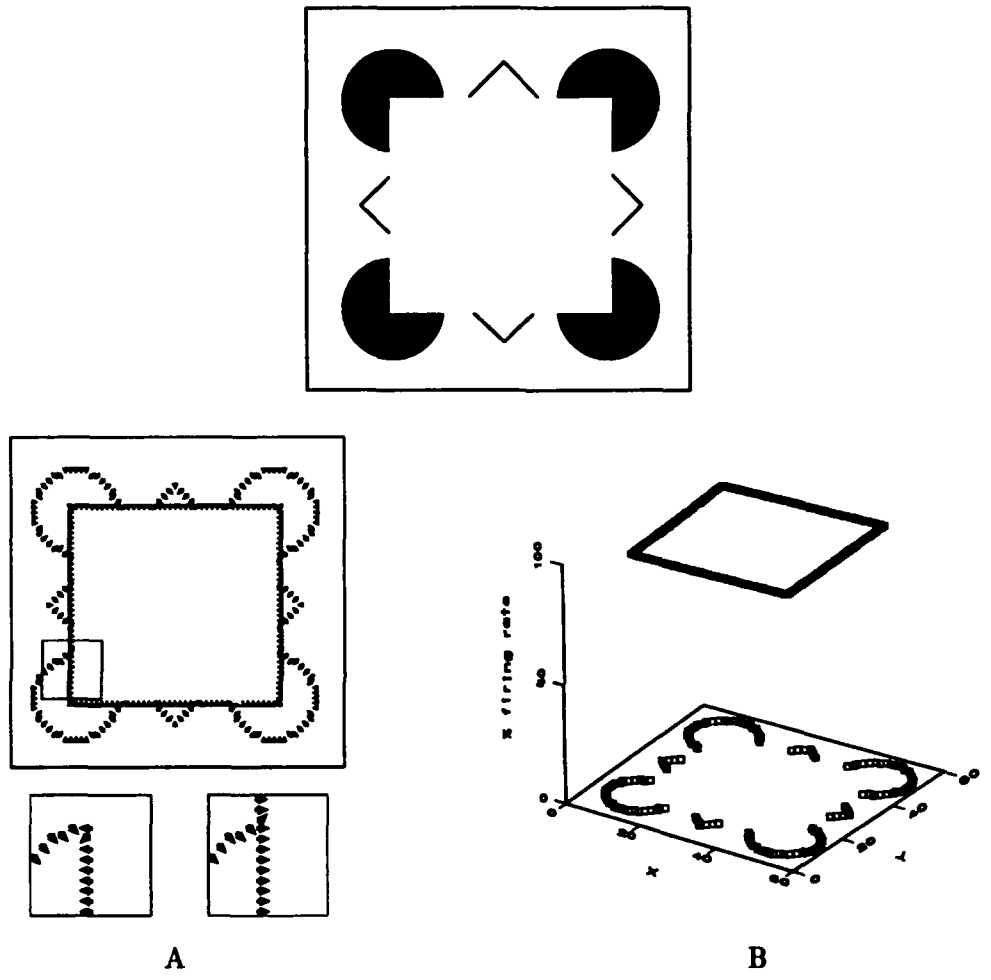



Figure 7: Upper panel shows a stimulus which is perceived by human subjects as an "illusory" square occluding four black discs and a wireframe square (rotated by 45°). A 64x64 discrete version of the stimulus was presented to the system. **A** Direction of figure determined by the system after three cycles. Insets show an enlarged view of a section of the output after the first and second cycle. For the first cycle the black disc "owns" the "L"–shaped segment of contour *(left)*. However, once the illusory square is generated the "ownership" flips to the illusory square *(right)*. **B** Activity in the foreground network (% maximum) demonstrating that the network correctly determines the relative depth for this illusory stimulus.

subprocesses, orientation and direction of figure, which are modeled at different levels of abstraction. Finally, we have shown that our integrated system of hybrid networks successfully discriminates objects and stratifies them in depth, while also accounting for several classes of human perceptual response.

A primary goal of both biological and artificial vision systems is object recognition. Whether the task is to find a deer in the forest or a screw on a conveyor belt, both systems must somehow recognize objects given a 2D image. Though Ullman (1989) points out that it is not *logically necessary* for object discrimination to take place before object recognition, it seems only reasonable that a visual system, whether biological or artificial, should use all processes at its disposal to generate meaningful representations of the scene. Our model suggests that by segmenting an image into its constituent objects, one is better able to identify that something is a "thing". This in turn should aid in the process of recognizing what kind of "thing" it is. Future models will extend the hybrid paradigm and integrate segmentation and recognition processes in order to create more complex models of visual perception and cognition.

## Acknowledgements

This work is supported by grants from The Office of Naval Research (N00014-90-J-1864), The Whitaker Foundation, and The McDonnell-Pew Program in Cognitive Neuroscience. P.S. is supported by an ONR-NDSEG Fellowship.

## References


[1] J. Aloimonos, I. Weiss, and A. Bandopadhay. Active vision. *Intl. Journal of Comp. Vision*, 1:333–356, 1987.

[2] R. Eckhorn, R. Bauer, W. Jordan, M. Brosch, W. Kruse, M. Munk, and H. Reitboeck. Coherent oscillations: A mechanism of feature linking in the visual cortex? *Biological Cybernetics*, 60:121–130, 1988.

[3] L.H. Finkel and P. Sajda. Object discrimination based on depth-from-occlusion. *Neural Computation*, (in press).

[4] R. B. Fisher. *From Objects to Surfaces.* John Wiley & Sons, New York, 1989.

[5] C. M. Gray and W. Singer. Neuronal oscillations in orientation columns of cat visual cortex. *Proceedings of the National Academy of Science USA*, 86:1698–1702, 1989.

[6] R. L. Gregory. Cognitive contours. *Nature*, 238:51–52, 1972.

[7] S. Grossberg and E. Mingolla. Neural dynamics of form perception: Boundary completion, illusory figures, and neon color spreading. *Psychology Review*, 92:173–211, 1985.

[8] A. Guzman. Decomposition of a visual scene into three-dimensional bodies. *Fall Joint Computer Conference*, 1968:291–304, 1968.

[9] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology*, 160:106–154, 1962.

[10] G. Kanizsa. *Organization in Vision.* Praeger, New York, 1979.

[11] K. Koffka. *Principles of Gestalt Psychology.* Harcourt, Brace, New York, 1935.

[12] S. Lehky and T. Sejnowski. Neural model of stereoacuity and depth interpolation based on distributed representation of stereo disparity. *Journal of Neuroscience*, 7:2281–2299, 1990.

[13] D. Marr. *Vision: A computational investigation into the Human Representation and Processing of Visual Information.* W. H. Freeman, San Francisco, 1982.

[14] D. Marr and E. Hildreth. Analysis of occluding contour. *Proceedings of the Royal Society of London (Biology)*, 197:441–475, 1977.

[15] D. Mumford, S. Kosslyn, L. Hillger, and R. Herrnstein. Discriminating figure from ground: The role of edge detection and region growing. *Proceedings of the National Academy of Science*, 84:7354–7358, 1987.

[16] K. Nakayama and S. Shimojo. Toward a neural understanding of visual surface representation. *Cold Spring Harbor Symposia on Quantitative Biology*, LV:911–924, 1990.

[17] E. Peterhans and R. von der Heydt. Mechanisms of contour perception in monkey visual cortex. II. Contours bridging gaps. *Journal of Neuroscience*, 9:1749–1763, 1989.

[18] G. F. Poggio, F. Gonzalez, and F. Krause. Stereoscopic mechanisms in monkey visual cortex: Binocular correlation and disparity selectivity. *Journal of Neuroscience*, 8:4531–4550, 1988.

[19] A. Rosenfeld. Computer vision. *Advances in Computers*, 27:265–308, 1988.

[20] P. Sajda and L. Finkel. NEXUS: A simulation environment for large-scale neural systems. *Simulation*, (in press).

[21] T. Sejnowski and G. Hinton. Separating figure from ground with a Boltzmann machine. In M. Arbib and A. Hanson, editors, *Vision, Brain and Cooperative Computation*, pages 703–724. MIT Press, Cambridge, MA, 1987.

[22] T. J. Sejnowski, C. Koch, and P. S. Churchland. Computational neuroscience. *Science*, 241:1299–1306, 1988.

[23] S. Ullman. Filling-in the gaps: The shape of subjective contours and a model for their generation. *Biological Cybernetics*, 25:1–6, 1977.

[24] S. Ullman. Aligning pictorial descriptions: An approach to object recognition. *Cognition*, 32:193–254, 1989.

[25] R. von der Heydt and E. Peterhans. Mechanisms of contour perception in monkey visual cortex. I. Lines of pattern discontinuity. *Journal of Neuroscience*, 9:1731–1748, 1989.